1. An apparatus in a processor for speculatively performing a return instruction, comprising:

first and second call/return stacks, for providing first and second return addresses, respectively;

a comparator, coupled to said first and second call/return stacks, for comparing said first and second return addresses; and

control logic, coupled to said comparator, for controlling the processor to branch to said first return address, said control logic subsequently controlling the processor to branch to said second return address if said comparator indicates said first and second return addresses do not match.

2. The apparatus of claim 1, wherein said second call/return stack is configured to provide said second return address in response to instruction decode logic decoding a return instruction.

3. The apparatus of claim 2, wherein said first call/return stack speculatively provides said first return address before decoding of said return instruction.

4. The apparatus of claim 3, wherein said first call/return stack speculatively provides said first return address in response to a fetch address, said fetch address selecting a cache line of an instruction cache.

5. The apparatus of claim 4, wherein said first call/return stack speculatively provides said first return address in response to said fetch address whether or not said return instruction is present in said cache line.

6. The apparatus of claim 1, further comprising:

a branch target address cache (BTAC), coupled to said first call/return stack, for caching a plurality of indications of whether a corresponding plurality of instructions previously executed by the processor are return instructions.

7. The apparatus of claim 6, wherein said first call/return stack provides said first return address in response to said BTAC providing one of said plurality of indications, wherein said one of said plurality of indications indicates that said corresponding instruction is a return instruction.

09845822.050401

8. The apparatus of claim 7, wherein said BTAC provides said one of said plurality of indications in response to an instruction cache fetch address.

9. The apparatus of claim 6, wherein said BTAC is further configured to cache a plurality of lengths of a corresponding plurality of call instructions previously executed by the processor.

10. The apparatus of claim 9, wherein said first return address comprises a sum of an instruction cache fetch address and one of said plurality of lengths provided by said BTAC.

11. The apparatus of claim 10, wherein said BTAC is further configured to cache a plurality of byte offsets within an instruction cache line of said corresponding plurality of call instructions, said byte offsets being within an instruction cache line selected by said fetch address.

12. The apparatus of claim 11, wherein said instruction cache line is selected by said fetch address.

13. The apparatus of claim 12, wherein said first return address comprises a sum of said instruction cache

fetch address and said one of said plurality of lengths and one of said plurality of byte offsets.

14. A microprocessor for predicting return instruction target addresses, comprising:

an instruction cache, for providing a line of instruction bytes selected by a fetch address, said fetch address provided on an address bus;

a speculative branch target address cache (BTAC), coupled to said address bus, for caching indications of previously executed return instructions, said speculative BTAC providing one of said indications to a speculative call/return stack in response to said fetch address whether or not a return instruction is present in said line of instruction bytes;

said speculative call/return stack coupled to said speculative BTAC, for providing a speculative return address to address selection logic in response to said one of said indications indicating one of said previously executed return instructions is potentially present in said line of instruction bytes; and

09849822.050401

said address selection logic configured to select said speculative return address as a subsequent fetch address for provision to said instruction cache.

15. The microprocessor of claim 14, further comprising:

address generation logic, coupled to said speculative call/return stack, for calculating said speculative return address for pushing onto said speculative call/return stack.

16. The microprocessor of claim 15, wherein said speculative BTAC is configured to cache indications of previously executed call instructions, said speculative BTAC providing one of said indications of one of said previously executed call instructions in response to said fetch address.

17. The microprocessor of claim 16, wherein said address generation logic calculates said speculative return address in response to said branch target address cache providing said one of said indications of one of said previously executed call instructions.

18. The microprocessor of claim 14, wherein said previously executed return instructions are x86 RET instructions.

19. A microprocessor for predicting return instruction target addresses, comprising:

an instruction cache, for generating a line of instruction bytes selected by a fetch address, said fetch address received from an address bus;

address selection logic, coupled to said address bus, for selecting said fetch address and providing said fetch address on said address bus;

a branch target address cache (BTAC), coupled to said address bus, for caching indications of previously executed return instructions and for providing one of said indications in response to said fetch address;

a first call/return stack, coupled to said BTAC, for providing a first return address to said address selection logic in response to said one of said indications;

decode logic, coupled to said instruction cache, for decoding said line of instruction bytes; and

a second call/return stack, coupled to said decode logic, for providing a second return address to said address selection logic in response to said decode logic indicating that a return instruction is present in said line of instruction bytes.

20. The microprocessor of claim 19, wherein said first call/return stack provides said first return address before said decode logic decodes said line of instruction bytes.

21. The microprocessor of claim 19, wherein said branch target address cache provides said one of said indications in response to said fetch address whether or not a return instruction is present in said line of instruction bytes.

22. The microprocessor of claim 19, wherein said first call/return stack provides said first return address in response to said one of said indications indicating said one of said previously executed return instructions is potentially present in said line of instruction bytes.

23. The microprocessor of claim 19, further comprising:

control logic, coupled to said BTAC, configured to control said address selection logic to select said first return address during a first period.

09846822.050401

24. The microprocessor of claim 23, further comprising:

a comparator, coupled to said first and second call/return stacks, for comparing said first and second return addresses.

25. The microprocessor of claim 24, wherein said control logic is further configured to control said address selection logic to select said second return address subsequent to controlling said address selection logic to select said first return address if said comparator indicates said first and second return addresses do not match.

26. The microprocessor of claim 19, wherein said second call/return stack provides said second return address subsequent to said first call/return stack providing said first return address.

09648822.050401

27. A method for speculatively branching a microprocessor to a target address of a return instruction, comprising:

generating a first target address by a first call/return stack;

branching to said first target address;

generating a second target address by a second call/return stack subsequent to said branching to said first target address;

comparing said first and second target addresses; and

branching to said second target address if said first and second target addresses do not match.

28. The method of claim 27, wherein said branching to said first target address comprises selecting said first target address and providing said first target address as a fetch address to an instruction cache in the microprocessor.

29. The method of claim 28, wherein said generating said first target address comprises said first call/return stack generating said first target address in response

09845822.050401

to a previous fetch address that was provided to said instruction cache.

30. The method of claim 29, wherein said generating said first target address is performed whether or not a return instruction is present in an instruction cache line selected by said fetch address.

31. The method of claim 29, further comprising:

decoding a return instruction present in a line of instruction bytes selected from said instruction cache by said fetch address, wherein said decoding said return instruction present in said line of instruction bytes is performed subsequent to said branching to said first target address.

32. The method of claim 31, wherein said generating said second target address comprises said second call/return stack generating said second target address in response to said decoding said return instruction present in said line of instruction bytes.

33. The method of claim 27, wherein said generating said first target address comprises popping said first target address off said first call/return stack.

09849522.050401

34. The method of claim 33, further comprising:

pushing said first target address onto said first call/return stack prior to said popping said first target address off said first call/return stack.

35. The method of claim 34, further comprising:

calculating said first target address prior to said pushing.

36. The method of claim 35, wherein said calculating said first target address comprises adding a cached length of a previously cached call instruction and a fetch address selecting an instruction cache line potentially including said previously executed call instruction.

37. The method of claim 36, wherein said generating said first target address comprises adding said fetch address, said cached length, and a cached offset of said call instruction within said instruction cache line.

38. The method of claim 34, wherein said pushing is performed in response to an instruction cache fetch address.

39. A microprocessor for predicting return instruction target addresses, comprising:

an instruction cache, for providing a line of instructions in response to a fetch address received on an address bus;

a multiplexer, having a plurality of inputs, configured to select one of said plurality of inputs for provision on said address bus as said fetch address to said instruction cache;

a speculative branch target address cache (BTAC), coupled to said address bus, for indicating a speculative presence of a return instruction in said line of instructions;

a speculative call/return stack, coupled to said speculative BTAC, for providing a speculative return address to a first of said plurality of multiplexer inputs in response to said speculative BTAC indicating said speculative presence of said return instruction;

09849822.050401

decode logic, configured to receive and decode said line of instructions;

a non-speculative call/return stack, coupled to said decode logic, for providing a non-speculative return address to a second of said plurality of multiplexer inputs in response to said decode logic indicating that said return instruction is actually present in said line of instructions; and

a comparator, coupled to said speculative and non-speculative call/return stacks, for comparing said speculative and non-speculative return addresses;

wherein said multiplexer selects said speculative return address in a first instance, and selects said non-speculative return address in a second instance subsequent to said first instance if said comparator indicates that said speculative and non-speculative return addresses do not match.

09849822.050401